snooping protocol

Last modified: Thursday, September 16, 2004

(snoop´ing pr ´t&-kōl´´) **(n.)** Also referred to as a *bus-snooping protocol*, a protocol for maintaining cache coherency in symmetric multiprocessing environments. In a snooping system, all caches on the bus monitor (or snoop) the bus to determine if they have a copy of the block of data that is requested on the bus. Every cache has a copy of the sharing status of every block of physical memory it has. Multiple copies of a document in a multiprocessing environment typically can be read without any coherence problems; however, a processor must have exclusive access to the bus in order to write.

There are two types of snooping protocol:

- **write-invalidate:** the processor that is writing data causes copies in the caches of all other processors in the system to be rendered invalid before it changes its local copy. The local machine does this by sending an invalidation signal over the bus, which causes all of the other caches to check for a copy of the invalidated file. Once the cache copies have been invalidated, the data on the local machine can be updated until another processor requests it.
- **write-update:** the processor that is writing the data broadcasts the new data over the bus (without issuing the invalidation signal). All caches that contain copies of the data are then updated. This scheme differs from write-invalidate in that it does not create only one local copy for writes.

•E-mail this definition to a colleague•

For internet.com pages about ***snooping protocol*** ***CLICK HERE***. Also check out

Related Categories

Buses

Caches

MENU
Home
Term of the Day
New Terms
Pronunciation
New Links
Quick Reference
Did You Know?
Search Tool
Tech Support
Webopedia Jobs
About Us
Link to Us
Advertising

Compare Prices: go
HardwareCentral

Talk To Us...
Submit a URL
Suggest a Term
Report an Error

internet.com
Developer
Downloads
International
Internet Lists
Internet News
Internet Resources
IT
Linux/Open Source
Personal Technology
Small Business

cache coherence

Last modified: Thursday, September 16, 2004

(cash c h r´&ns) (n.) A protocol for managing the caches of a multiprocessor system so that no data is lost or overwritten before the data is transferred from a cache to the target memory. When two or more computer processors work together on a single program, known as multiprocessing, each processor may have its own memory cache that is separate from the larger RAM that the individual processors will access. A memory cache, sometimes called a *cache store* or RAM cache, is a portion of memory made of high-speed static RAM (SRAM) instead of the slower and cheaper dynamic RAM (DRAM) used for main memory. Memory caching is effective because most programs access the same data or instructions over and over. By keeping as much of this information as possible in SRAM, the computer avoids accessing the slower DRAM.

When multiple processors with separate caches share a common memory, it is necessary to keep the caches in a state of coherence by ensuring that any shared operand that is changed in any cache is changed throughout the entire system. This is done in either of two ways: through a directory-based or a snooping system. In a directory-based system, the data being shared is placed in a common directory that maintains the coherence between caches. The directory acts as a filter through which the processor must ask permission to load an entry from the primary memory to its cache. When an entry is changed the directory either updates or invalidates the other caches with that entry. In a snooping system, all caches on the bus monitor (or snoop) the bus to determine if they have a copy of the block of data that is requested on the bus. Every cache has a copy of the sharing status of every block of physical memory it has.